Asymmetric key cryptography

[Slide courtesy: Cryptography and network security by Behrouz Fourozan]

Introduction

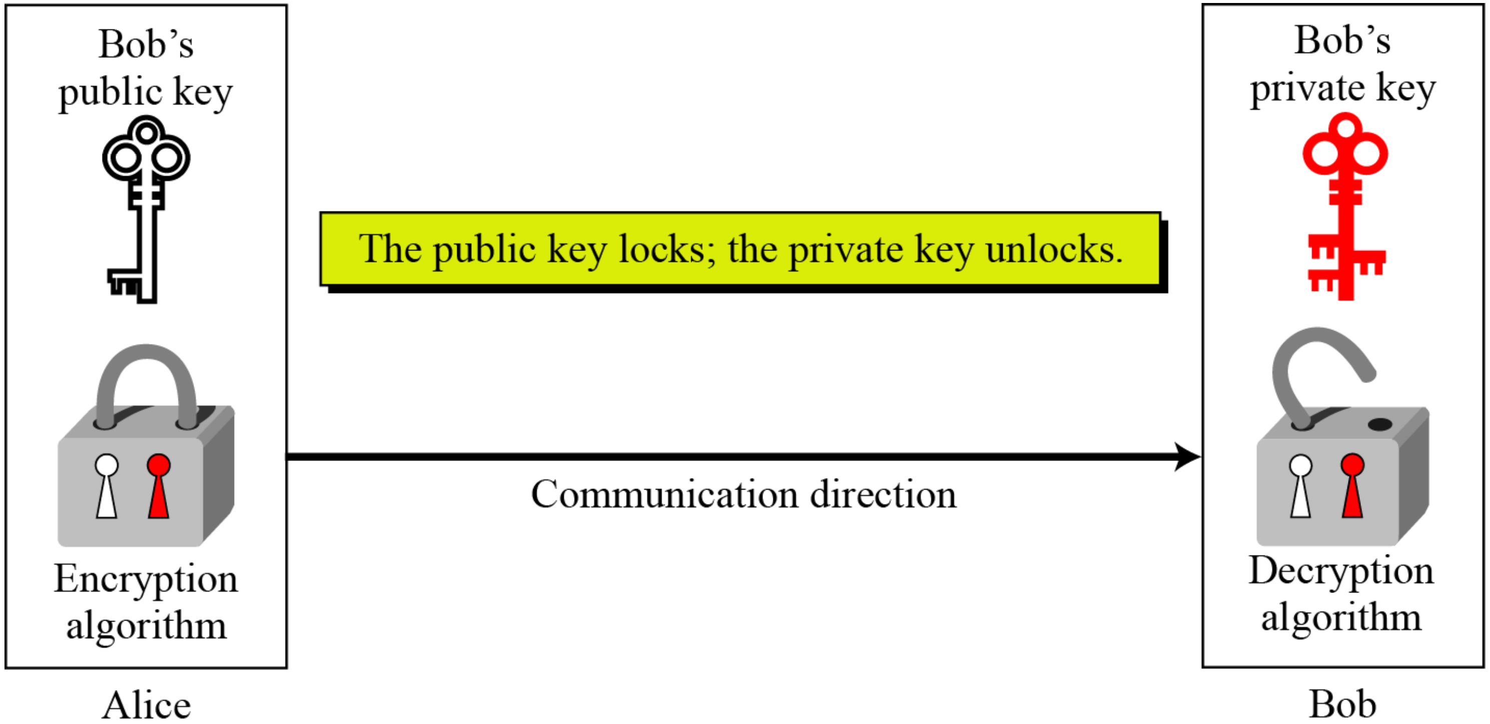
- Symmetric and asymmetric-key cryptography will exist in parallel and continue to serve the community.
- They are complements of each other
  - The advantages of one can compensate for the disadvantages of the other.
- Symmetric-key cryptography is based on sharing secrecy
- Asymmetric-key cryptography is based on personal secrecy.

Need for Both

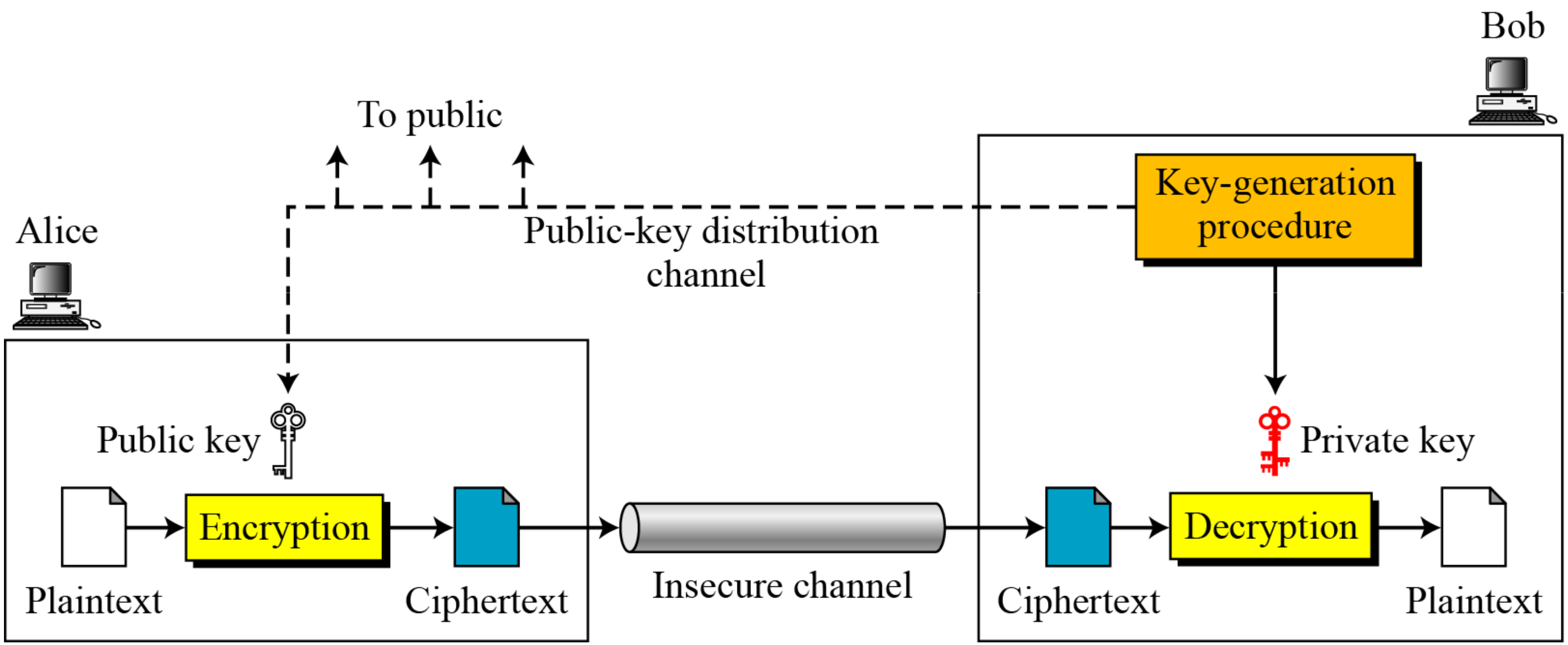
- There is a very important fact that is sometimes misunderstood
- The advent of asymmetric-key cryptography does not eliminate the need for symmetric-key cryptography.

Keys

- Asymmetric key cryptography uses two separate keys
  - one private and one public.



General Idea



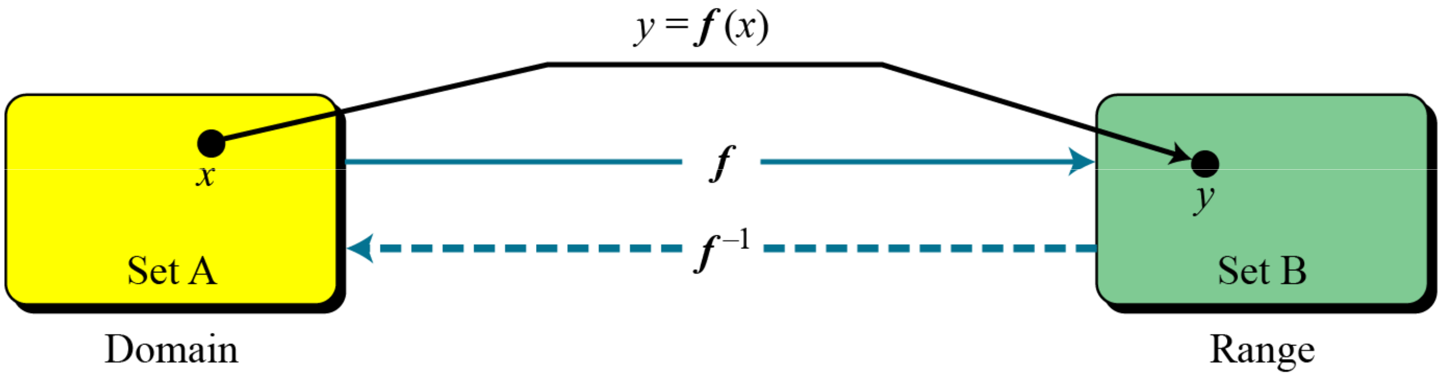
General Idea...

- Plaintext/Ciphertext
  - Unlike in symmetric-key cryptography, plaintext and ciphertext are treated as integers in asymmetric-key cryptography.

$$\boxed{C = f\,(K_{public},\ P) \qquad P = g(K_{private},\ C)}$$

Trapdoor One-Way Function

- The main idea behind asymmetric-key cryptography



Trapdoor One-Way Function...

- One-Way Function (OWF)

1. f is easy to compute.
2. f^{-1} is difficult to compute.

- Trapdoor One-Way Function (TOWF)

3. Given y and a trapdoor, x can be computed easily.

Trapdoor One-Way Function…

- Example
  - When n is large, $n = p \times q$ is a one-way function. Given p and q , it is always easy to calculate n ; given n, it is very difficult to compute p and q. This is the factorization problem.
- Example
  - When n is large, the function $y = x^k \bmod n$ is a trapdoor one-way function. Given x, k, and n, it is easy to calculate y. Given y, k, and n, it is very difficult to calculate x. This is the discrete logarithm problem. However, if we know the trapdoor, k′ such that $k \times k' = 1 \bmod \Phi(n)$, we can use $x = y^{k'} \bmod n$ to find x.

Merkle-Hellman Knapsack Cryptosystem

- Definition
  - $a = [a_1, a_2, \ldots, a_k]$ and $x = [x_1, x_2, \ldots, x_k]$.

$$\boxed{s = knapsackSum\ (a, x)\ =\ x_1a_1 + x_2a_2 + \cdots +\ x_ka_k}$$

  - Given a and x, it is easy to calculate s. However, given s and a it is difficult to find x.
- Superincreasing Tuple
-

$$\boxed{a_i \geq a_1 + a_2 + \ldots + a_{i-1}}$$

Merkle-Hellman Knapsack Cryptosystem...

Algorithm 10.1 *knapsacksum and inv_knapsackSum for a superincreasing k-tuple*

knapsackSum (x [1 … k], a [1 … k]) { $s \leftarrow 0$ for (i = 1 to k) { $s \leftarrow s + a_i \times x_i$ } return s }	**inv_knapsackSum** (s, a [1 … k]) { for (i = k down to 1) { if $s \geq a_i$ { $x_i \leftarrow 1$ $s \leftarrow s - a_i$ } else $x_i \leftarrow 0$ } return x [1 … k] }

Merkle-Hellman Knapsack Cryptosystem…

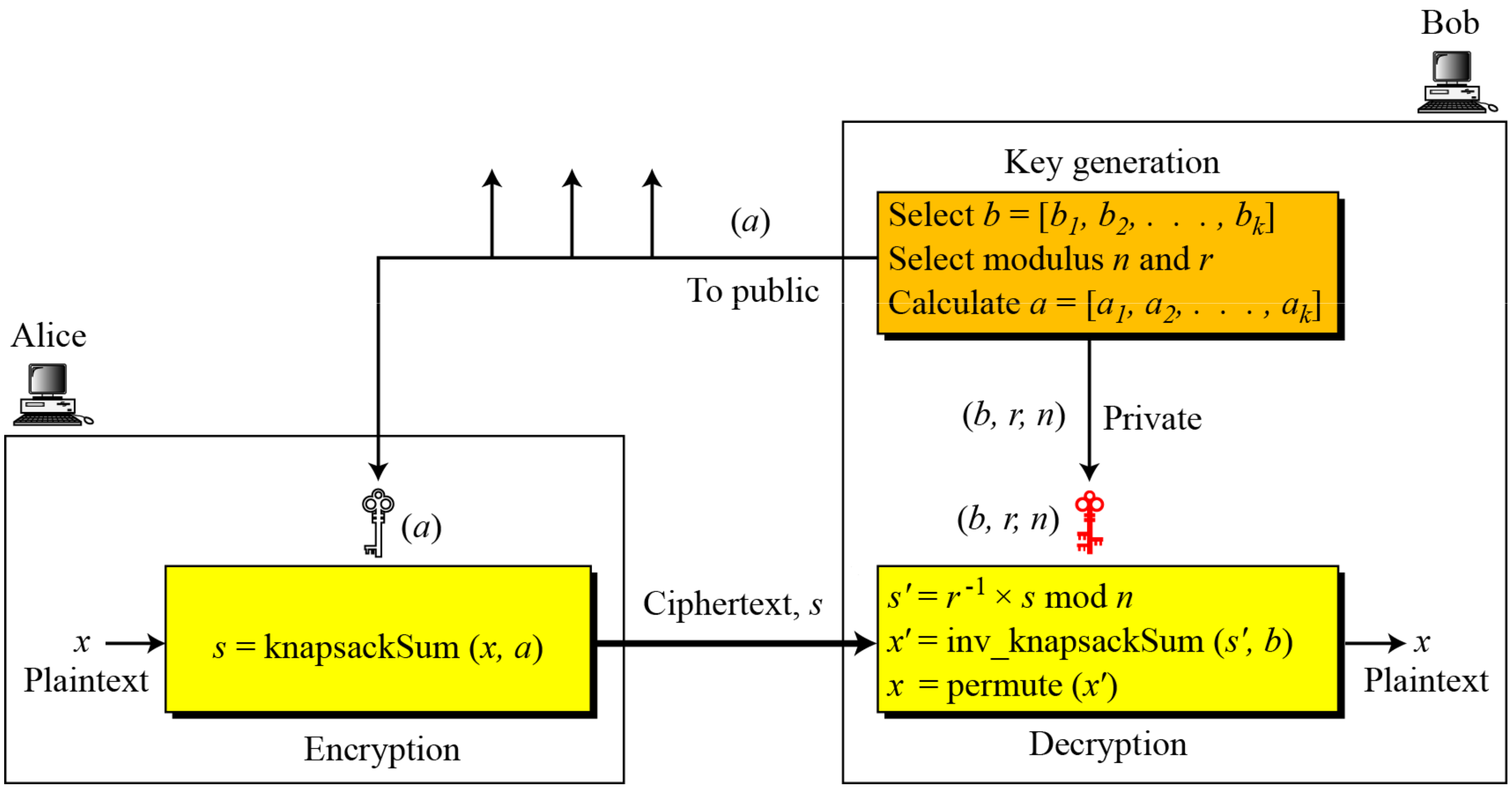
- Example
  - As a very trivial example, assume that a = [17, 25, 46, 94, 201,400] and s = 272 are given. Table 10.1 shows how the tuple x is found using inv_knapsackSum routine in Algorithm 10.1. In this case x = [0, 1, 1, 0, 1, 0], which means that 25, 46, and 201 are in the knapsack.

Table 10.1 *Values of i, a_i, s, and x_i in Example 10.3*

i	a_i	s	$s \geq a_i$	x_i	$s \leftarrow s - a_i \times x_i$
6	400	272	false	$x_6 = 0$	272
5	201	272	true	$x_5 = 1$	71
4	94	71	false	$x_4 = 0$	71
3	46	71	true	$x_3 = 1$	25
2	25	25	true	$x_2 = 1$	0
1	17	0	false	$x_1 = 0$	0

Merkle-Hellman Knapsack Cryptosystem…

- Secret Communication with Knapsacks.



Merkle-Hellman Knapsack Cryptosystem…

1. Key generation:
   a. Bob creates the superincreasing tuple $b = [7, 11, 19, 39, 79, 157, 313]$.
   b. Bob chooses the modulus $n = 900$ and $r = 37$, and [4 2 5 3 1 7 6] as permutation table.
   c. Bob now calculates the tuple $t = [259, 407, 703, 543, 223, 409, 781]$.
   d. Bob calculates the tuple a = permute $(t) = [543, 407, 223, 703, 259, 781, 409]$.
   e. Bob publicly announces a; he keeps n, r, and b secret.

2. Suppose Alice wants to send a single character "g" to Bob.
   a. She uses the 7-bit ASCII representation of "g", $(1100111)_2$, and creates the tuple $x = [1, 1, 0, 0, 1, 1, 1]$. This is the plaintext.
   b. Alice calculates $s = knapsackSum\ (a, x) = 2165$. This is the ciphertext sent to Bob.

3. Bob can decrypt the ciphertext, $s = 2165$.
   a. Bob calculates $s' = s \times r^{-1} \bmod n = 2165 \times 37^{-1} \bmod 900 = 527$.
   b. Bob calculates $x' = Inv\_knapsackSum\ (s', b) = [1, 1, 0, 1, 0, 1, 1]$.
   c. Bob calculates $x = permute\ (x') = [1, 1, 0, 0, 1, 1, 1]$. He interprets the string $(1100111)_2$ as the character "g".

Merkle-Hellman Knapsack Cryptosystem…

- Exercise

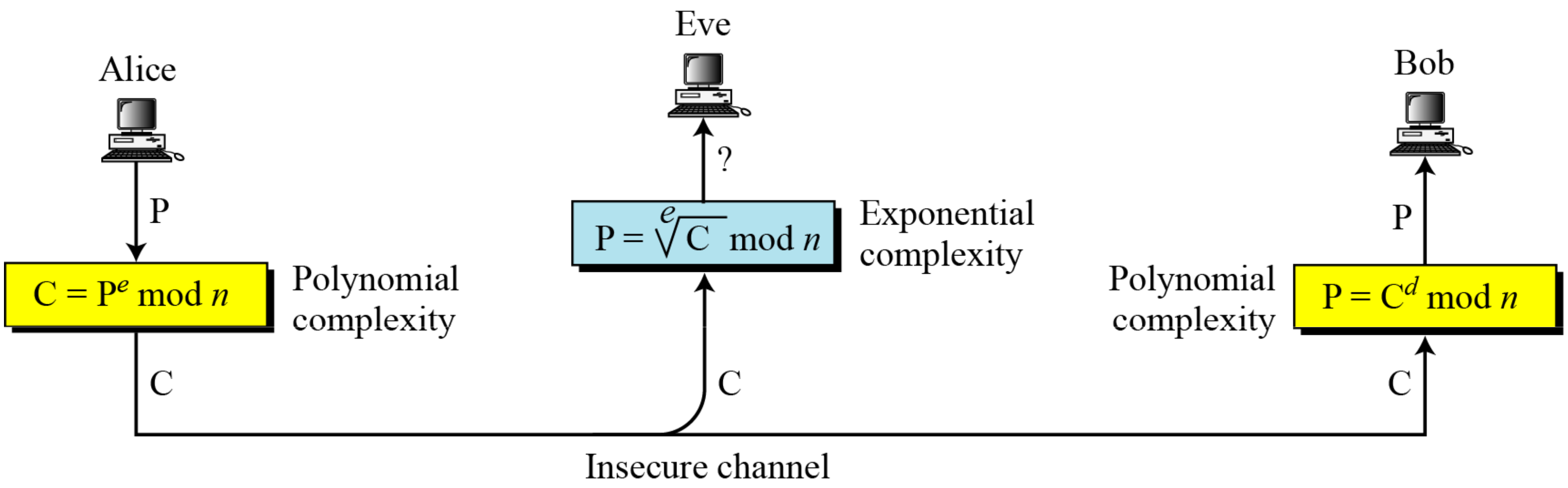
Given the superincreasing tuple b=[7,11,23,43,87,173,357], r=41 and modulus n=1001, encrypt and decrypt the letter 'a' using the Merkle-Hellman knapsack cryptosystem.

Use [7 6 5 1 2 3 4] as the permutation table.

RSA CRYPTOSYSTEM

- The most common public-key algorithm is the RSA cryptosystem, named for its inventors (Rivest, Shamir, and Adleman).

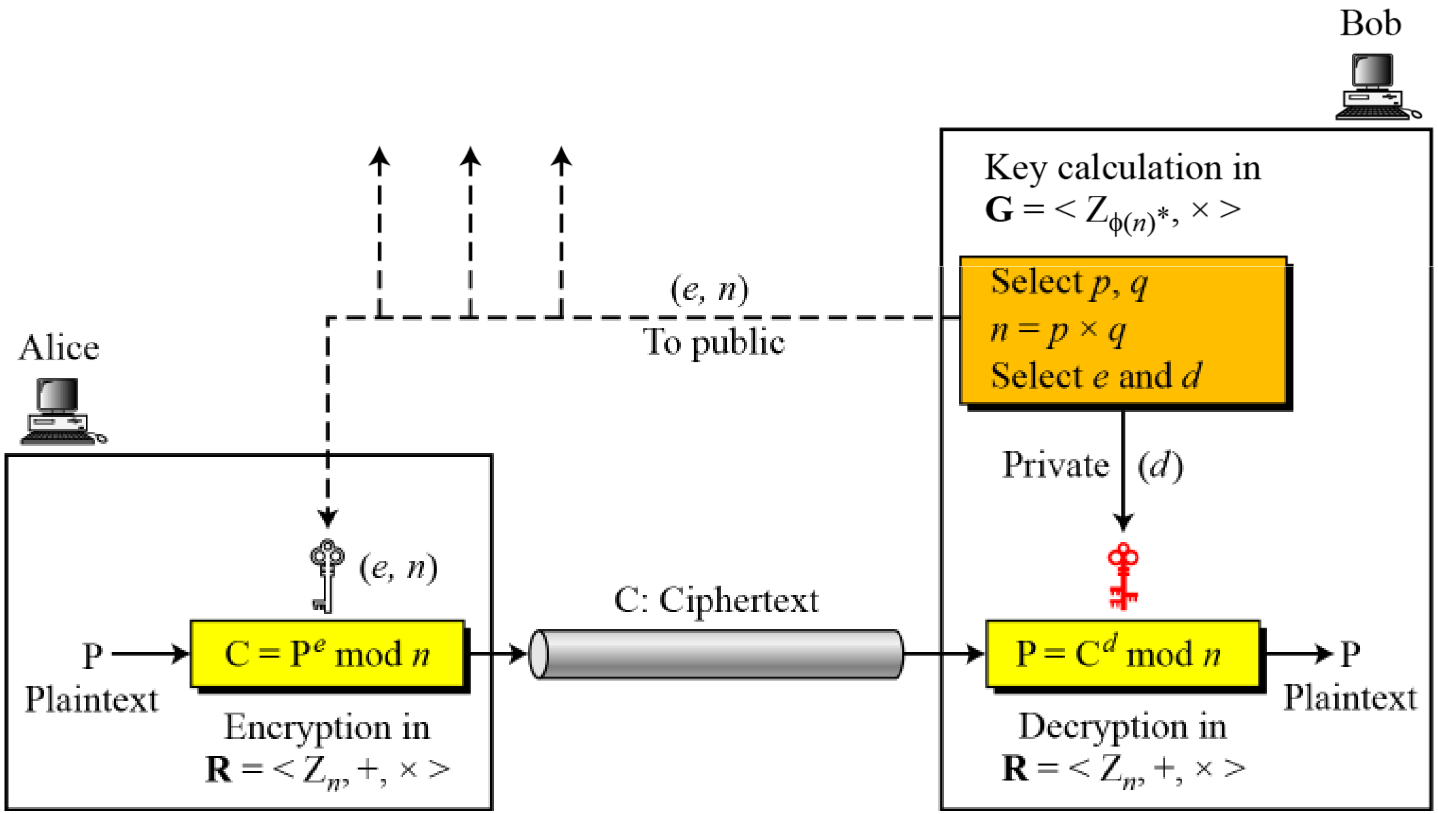
RSA CRYPTOSYSTEM…



RSA uses modular exponentiation for encryption/decryption; To attack it, Eve needs to calculate $\sqrt[e]{C} \bmod n$.

RSA CRYPTOSYSTEM…

- Encryption, decryption, and key generation in RSA



RSA CRYPTOSYSTEM…

- Two Algebraic Structures

Encryption/Decryption Ring: $R = <Z_n, +, \times>$

Key-Generation Group: $G = <Z_{\phi(n)}{}^*, \times>$

RSA uses two algebraic structures:
a public ring $R = <Z_n, +, \times>$ and a private group $G = <Z_{\phi(n)}{}^*, \times>$.

In RSA, the tuple (e, n) is the public key; the integer d is the private key.

RSA CRYPTOSYSTEM…

Algorithm 10.2 *RSA Key Generation*

```
RSA_Key_Generation
{
    Select two large primes p and q such that p ≠ q.
    n ← p × q
    φ(n) ← (p − 1) × (q − 1)
    Select e such that 1 < e < φ(n) and e is coprime to φ(n)
    d ← e^(−1) mod φ(n)                  // d is inverse of e modulo φ(n)
    Public_key ← (e, n)                   // To be announced publicly
    Private_key ← d                       // To be kept secret
    return Public_key and Private_key
}
```

RSA CRYPTOSYSTEM…

Encryption

Algorithm 10.3 *RSA encryption*

```
RSA_Encryption (P, e, n)                    // P is the plaintext in Z_n and P < n
{
    C ← Fast_Exponentiation (P, e, n)       // Calculation of (P^e mod n)
    return C
}
```

In RSA, p and q must be at least 512 bits; n must be at least 1024 bits.

RSA CRYPTOSYSTEM…

Decryption

Algorithm 10.4 *RSA decryption*

```
RSA_Decryption (C, d, n)                    //C is the ciphertext in Z_n
{
    P ← Fast_Exponentiation (C, d, n)       // Calculation of (C^d mod n)
    return P
}
```

RSA CRYPTOSYSTEM…

Can you give a proof of RSA?

RSA CRYPTOSYSTEM…

- Proof of RSA

If $n = p \times q$, $a < n$, and k is an integer, then $a^{k \times \phi(n)+1} \equiv a \pmod{n}$.

$$
\begin{array}{ll}
P_1 = C^d \bmod n = (P^e \bmod n)^d \bmod n = P^{ed} \bmod n & \\
ed = k\phi(n) + 1 & \text{// } d \text{ and } e \text{ are inverses modulo } \phi(n) \\
P_1 = P^{ed} \bmod n \;\rightarrow\; P_1 = P^{k\phi(n)+1} \bmod n & \\
P_1 = P^{k\phi(n)+1} \bmod n = \;\; P \bmod n & \text{// Euler's theorem (second version)}
\end{array}
$$

Some Trivial Examples

- Example
  - Bob chooses 7 and 11 as p and q and calculates $n = 77$. The value of $\Phi(n) = (7 - 1)(11 - 1)$ or 60. Now he chooses two exponents, e and d, from $Z_{60}*$. If he chooses e to be 13, then d is 37. Note that $e \times d \bmod 60 = 1$ (they are inverses of each other). Now imagine that Alice wants to send the plaintext 5 to Bob. She uses the public exponent 13 to encrypt 5.

Plaintext: 5	$C = 5^{13} = 26 \bmod 77$	Ciphertext: 26

  - Bob receives the ciphertext 26 and uses the private key 37 to decipher the ciphertext:

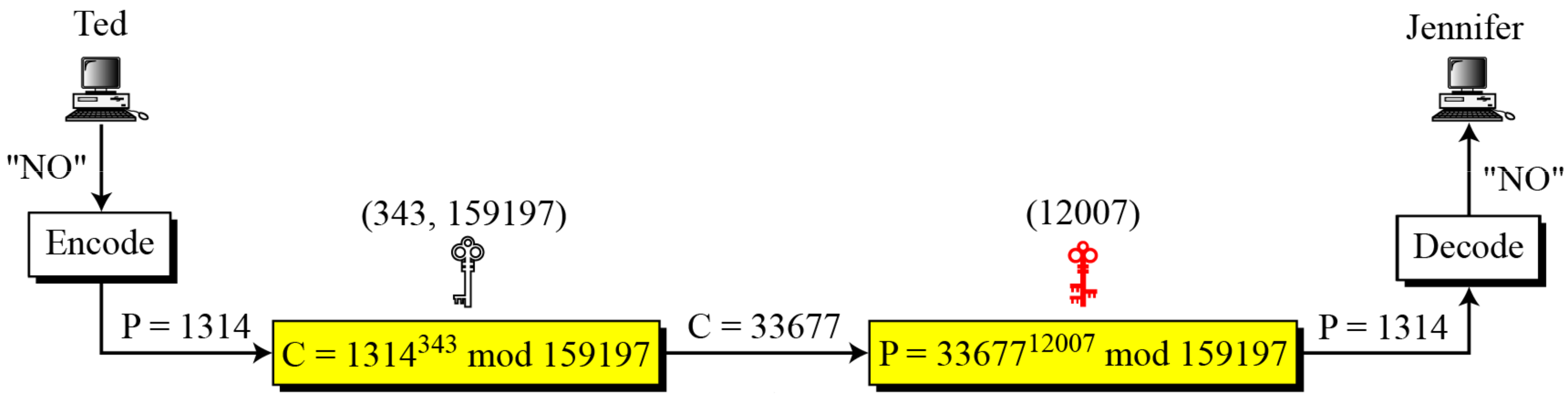
Ciphertext: 26	$P = 26^{37} = 5 \bmod 77$	Plaintext: 5

Some Trivial Examples…

Jennifer creates a pair of keys for herself. She chooses p = 397 and q = 401. She calculates n = 159197. She then calculates $\Phi(n)$ = 158400. She then chooses e = 343 and d = 12007. Show how Ted can send a message to Jennifer if he knows e and n.

Suppose Ted wants to send the message "NO" to Jennifer. He changes each character to a number (from 00 to 25), with each character coded as two digits. He then concatenates the two coded characters and gets a four-digit number. The plaintext is 1314.

Some Trivial Examples…



A realistic example

- A more realistic example
- We choose a 512-bit p and q, calculate n and Φ(n), then choose e and test for relative primeness with Φ(n). We then calculate d. Finally, we show the results of encryption and decryption. The integer p is a 159-digit number.

$p =$	9613034531358350457419158128061542790930984559499621582258315087964794045505647063849125716018034750312098666606492420191808780667421096063354219926661209

$q =$	12060191957231446918276794204450896001555925054637033936061798321731482148483764659215389453209175225273226830107120695604602513887145524969000359660045617

A realistic example...

- The modulus n = p × q. It has 309 digits.

$n =$	115935041739676149688925098646158875237714573754541447754855261376 147885408326350817276878815968325168468849300625485764111250162414 552339182927162507656772727460097082714127730434960500556347274566 628060099924037102991424472292215772798531727033839381334692684137 327622000966676671831831088373420823444370953

- Φ(n) = (p − 1)(q − 1) has 309 digits.

$\phi(n) =$	115935041739676149688925098646158875237714573754541447754855261376 147885408326350817276878815968325168468849300625485764111250162414 552339182927162507656751054233608492916752034482627988117554787657 013923444405716989581728196098226361075467211864612171359107358640 614008885170265377277264467341066243857664128

A realistic example...

- Bob chooses e = 35535 and tests it to make sure it is relatively prime with Φ(n). He then finds the inverse of e modulo Φ(n) and calls it d.

$e =$	35535
$d =$	58008302860037763936093661289677917594669062089650962180422866111380593852822358731706286910030021710859044338402170729869087600611530620252495988444804756824096624708148581713046324064407770483313401085094738529564507193677406119732655742423721761767462077637164207600337085333288532144708859551366702948 31

A realistic example…

- Example
  - Alice wants to send the message “THIS IS A TEST”, which can be changed to a numeric value using the 00−26 encoding scheme (26 is the space character).

P =	1907081826081826002619041819

  - The ciphertext calculated by Alice is C = P^e, which is

C =	47530912364622682720636555061054518094237179607049171652323924305445296061319932856661784341835911415119741125200568297979457173603610127821884789274156609048002350719071527718591497518846588863210114835410336165789846796838676373376577746562507928052114814184404814184430812773059004692874248559166462108656

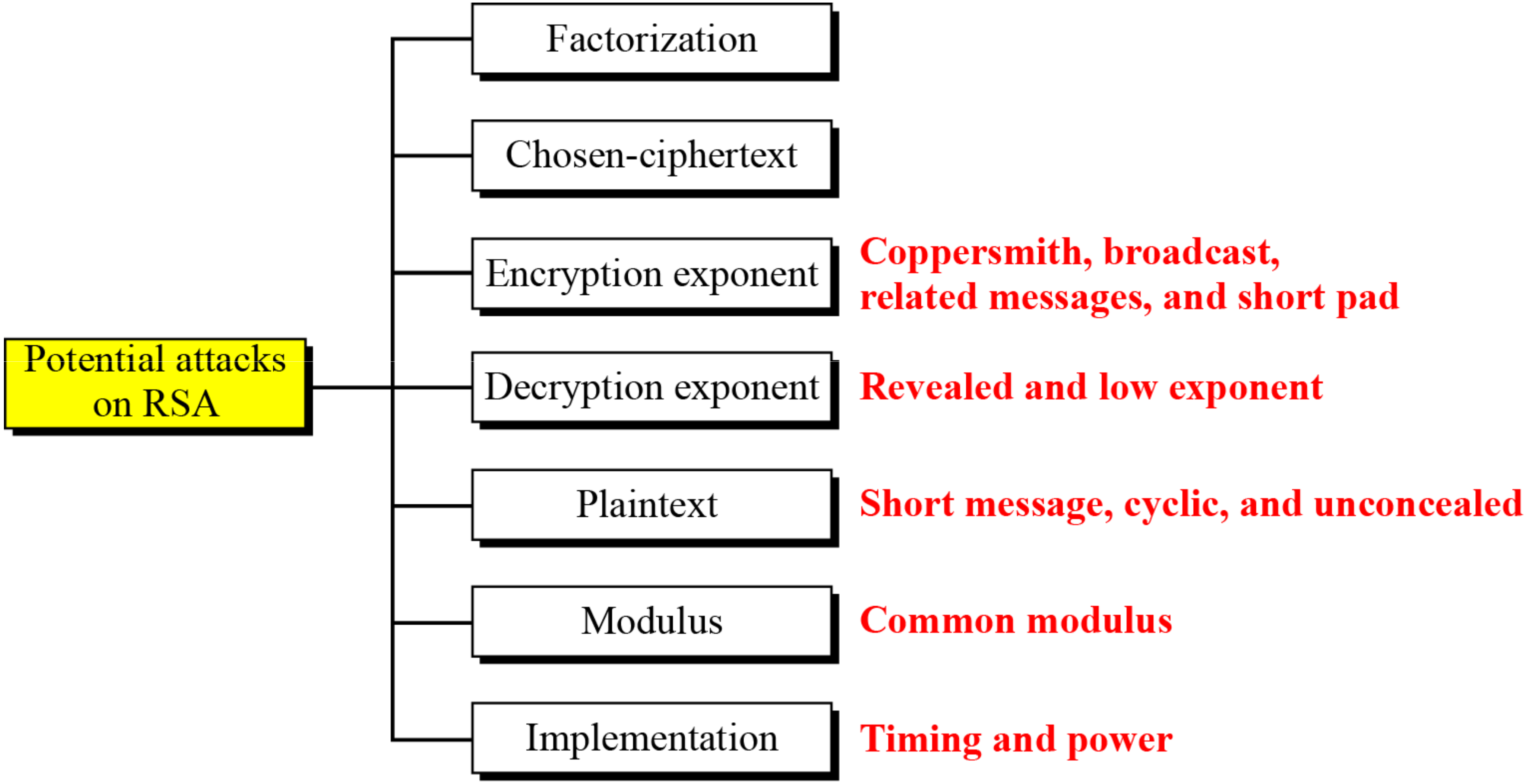
A realistic example...

- Bob can recover the plaintext from the ciphertext using $P = C^d$, which is

P =	1907081826081826002619041819

- The recovered plaintext is “THIS IS A TEST” after decoding.

Attacks on RSA

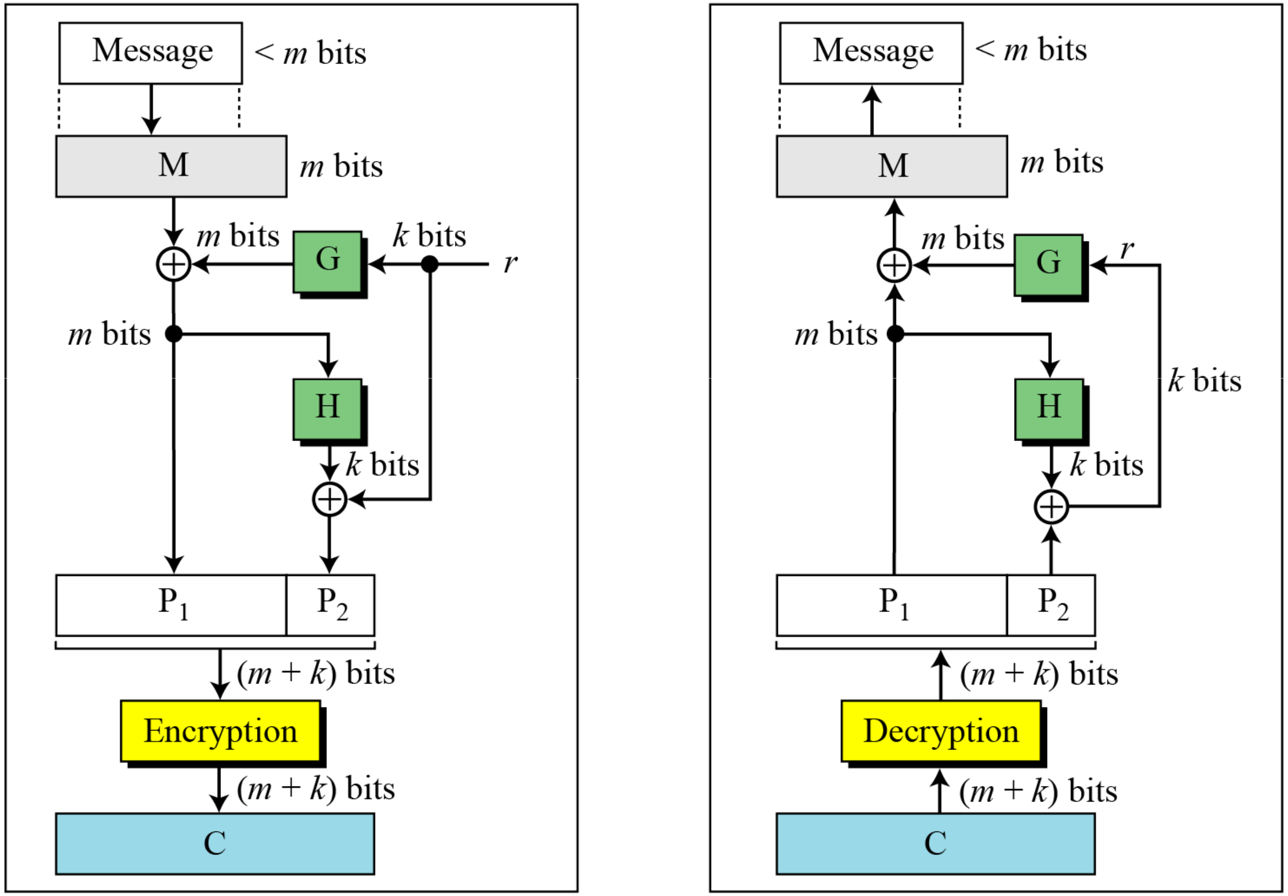


OAEP:Optimal Asymmetric Encryption Padding

M: Padded message
r: One-time random number

P: Plaintext ($P_1 \| P_2$)
C: Ciphertext

G: Public function (k-bit to m-bit)
H: Public function (m-bit to k-bit)



Sender Receiver

QUADRATIC CONGRUENCE

- In cryptography, we also need to discuss quadratic congruence that is, equations of the form
  $a_2x^2 + a_1x + a_0 \equiv 0 \pmod{n}$.
- We limit our discussion to quadratic equations in which $a_2 = 1$ and $a_1 = 0$, that is equations of the form

$$\boxed{x^2 \equiv a \pmod{n}.}$$

Quadratic Congruence Modulo a Prime

- Example
  - The equation $x^2 \equiv 3 \pmod{11}$ has two solutions,
  - $x \equiv 5 \pmod{11}$ and $x \equiv -5 \pmod{11}$.
  - But note that $-5 \equiv 6 \pmod{11}$, so the solutions are actually 5 and 6.
- Example
  - The equation $x^2 \equiv 2 \pmod{11}$ has no solution. No integer x can be found such that its square is 2 mod 11.

Quadratic Congruence Modulo a Prime…

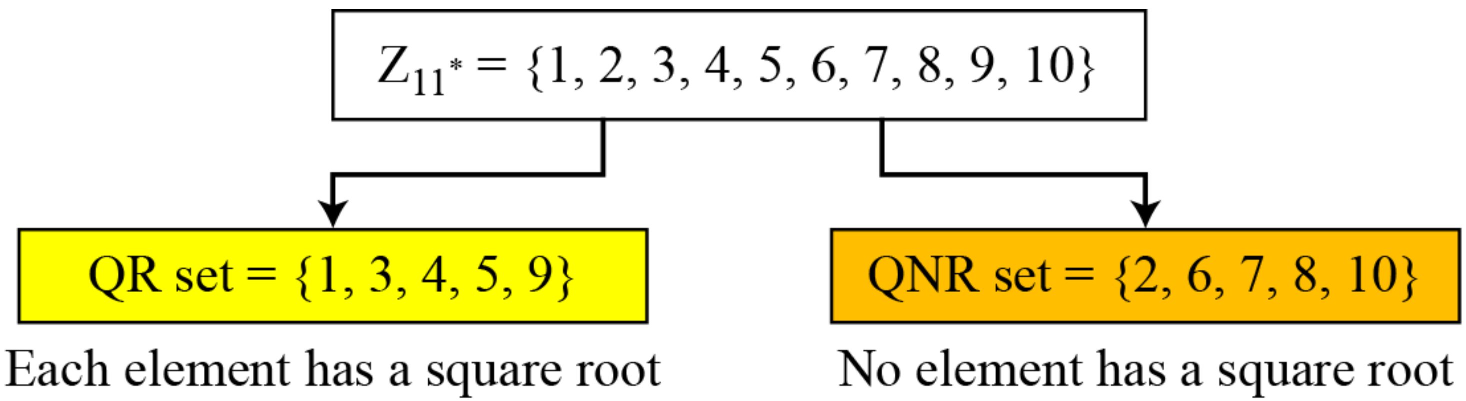
- Quadratic Residues and Nonresidue
  - In the equation $x^2 \equiv a \pmod{p}$, a is called a quadratic residue (QR) if the equation has two solutions;
  - a is called quadratic nonresidue (QNR) if the equation has no solutions.

Quadratic Congruence Modulo a Prime…

- Example
  - How many QRs in Z_{11}* ?
  - How many QNRs in Z_{11}* ?

Quadratic Congruence Modulo a Prime…

- Example
  - There are 10 elements in Z_{11}^*.
  - Exactly five of them are quadratic residues and five of them are nonresidues.
  - In other words, Z_{11}^* is divided into two separate sets, QR and QNR, as shown in Figure.



Quadratic Congruence Modulo a Prime…

- Euler's Criterian
  - If $a^{(p-1)/2} \equiv 1 \pmod{p}$, a is a quadratic residue modulo p.
  - If $a^{(p-1)/2} \equiv -1 \pmod{p}$, a is a quadratic nonresidue modulo p.
- Example
  - Find out if 14 or 16 is a QR in Z_{23}^*

Quadratic Congruence Modulo a Prime…

- Euler's Criterian
  - If $a^{(p-1)/2} \equiv 1 \pmod{p}$, a is a quadratic residue modulo p.
  - If $a^{(p-1)/2} \equiv -1 \pmod{p}$, a is a quadratic nonresidue modulo p.
- Example
  - Find out if 14 or 16 is a QR in $Z_{23}{}^{*}$

$14^{(23-1)/2} \bmod 23 \rightarrow 22 \bmod 23 \rightarrow -1 \bmod 23$ nonresidue

$16^{(23-1)/2} \bmod 23 \rightarrow 16^{11} \bmod 23 \rightarrow 1 \bmod 23$ residue

Quadratic Congruence Modulo a Prime…

- Special case
  - Special Case: p = 4k + 3

$$x \equiv a^{(p+1)/4} \pmod{p} \qquad \text{and} \qquad x \equiv -a^{(p+1)/4} \pmod{p}$$

Quadratic Congruence Modulo a Prime…

- Examples
  - Solve the following quadratic congruences

a. $x^2 \equiv 3 \pmod{23}$

b. $x^2 \equiv 2 \pmod{11}$

c. $x^2 \equiv 7 \pmod{19}$

Quadratic Congruence Modulo a Prime…

- Examples
  - Solve the following quadratic congruences

$$\begin{aligned} &\text{a.} \quad x^2 \equiv 3 \pmod{23} \\ &\text{b.} \quad x^2 \equiv 2 \pmod{11} \\ &\text{c.} \quad x^2 \equiv 7 \pmod{19} \end{aligned}$$

- Solution

a. $x \equiv \pm 16 \pmod{23}$ $\sqrt{3} \equiv \pm 16 \pmod{23}$.

b. There is no solution for $\sqrt{2}$ in Z_{11}.

c. $x \equiv \pm 11 \pmod{19}$. $\sqrt{7} \equiv \pm 11 \pmod{19}$.

Quadratic Congruence Modulo a Composite

$$\boxed{\begin{array}{l} x^2 \equiv a \bmod (n) \\ n = p_1 \times p_2 \times \, . \; . \; . \times p_k \end{array}} \longrightarrow \boxed{\begin{array}{c} x^2 \equiv a_1 \pmod{p_1} \\ x^2 \equiv a_2 \pmod{p_1} \\ \bullet\bullet\bullet \\ x^2 \equiv a_k \pmod{p_k} \end{array}} \longrightarrow \boxed{\begin{array}{c} x_1 \equiv \pm\, b_1 \pmod{p_1} \\ x_2 \equiv \pm\, b_2 \pmod{p_1} \\ \bullet\bullet\bullet \\ x_k \equiv \pm\, b_k \pmod{p_k} \end{array}}$$

Quadratic Congruence Modulo a Composite…

- Example
  - Assume that $x^2 \equiv 36 \pmod{77}$. We know that $77 = 7 \times 11$. We can write

$$x^2 \equiv 36 \pmod{7} \equiv 1 \pmod{7} \qquad \text{and} \qquad x^2 \equiv 36 \pmod{11} \equiv 3 \pmod{11}$$

  - The answers are $x \equiv +1 \pmod{7}$, $x \equiv -1 \pmod{7}$, $x \equiv +5 \pmod{11}$, and $x \equiv -5 \pmod{11}$. Now we can make four sets of equations out of these:

$$\begin{array}{lll}
\textbf{Set 1:} & x \equiv +1 \pmod{7} & x \equiv +5 \pmod{11} \\
\textbf{Set 2:} & x \equiv +1 \pmod{7} & x \equiv -5 \pmod{11} \\
\textbf{Set 3:} & x \equiv -1 \pmod{7} & x \equiv +5 \pmod{11} \\
\textbf{Set 4:} & x \equiv -1 \pmod{7} & x \equiv -5 \pmod{11}
\end{array}$$

  - The answers are $x = \pm 6$ and ± 27.

Quadratic Congruence Modulo a Composite…

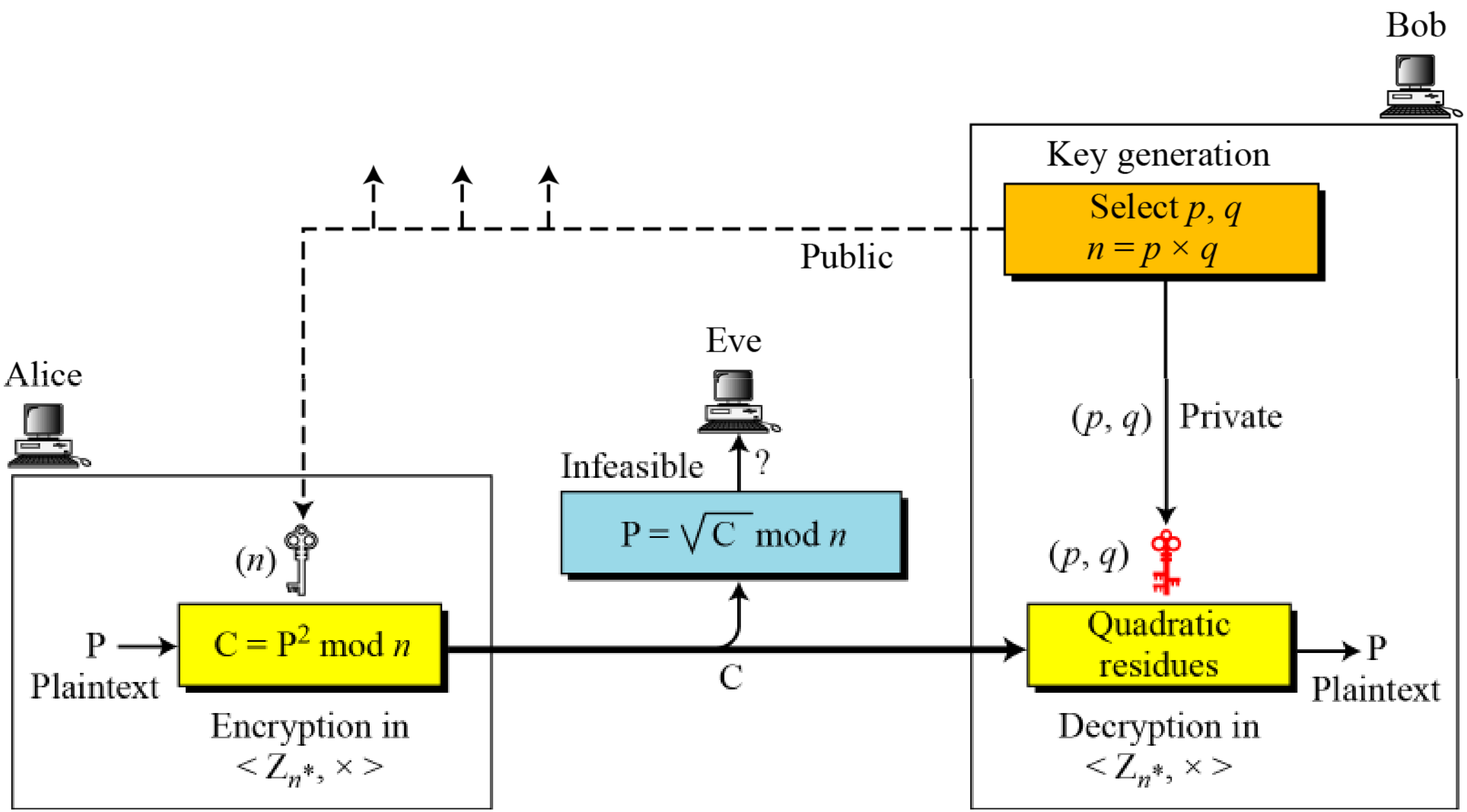
- Complexity
  - How hard is it to solve a quadratic congruence modulo a composite?
  - The main task is the factorization of the modulus.
  - Therefore, the complexity of solving a quadratic congruence modulo a composite is the same as factorizing a composite integer.
  - If *n* is very large, factorization is infeasible.

Solving a quadratic congruence modulo a composite is as hard as factorization of the modulus.

Rabin Cryptosystem

- The Rabin cryptosystem can be thought of as an RSA cryptosystem in which the value of e and d are fixed.
- The encryption is $C \equiv P^2 \pmod{n}$ and the decryption is $P \equiv C^{1/2} \pmod{n}$.
- The value of p and q are private

Rabin Cryptosystem…



Rabin Cryptosystem…

Algorithm 10.6 *Key generation for Rabin cryptosystem*

```
Rabin_Key_Generation
{
    Choose two large primes p and q in the form 4k + 3 and p ≠ q.
    n ← p × q
    Public_key ← n                          // To be announced publicly
    Private_key ← (q, n)                    // To be kept secret
    return Public_key and Private_key
}
```

Rabin Cryptosystem...

Algorithm 10.7 *Encryption in Rabin cryptosystem*

```
Rabin_Encryption (n, P)        // n is the public key; P is the ciphertext from Z_n*
{
   C ← P^2 mod n               // C is the ciphertext
   return C
}
```

Algorithm 10.8 *Decryption in Rabin cryptosystem*

```
Rabin_Decryption (p, q, C)     // C is the ciphertext; p and q are private keys
{
   a1 ← +(C^((p+1)/4)) mod p
   a2 ← −(C^((p+1)/4)) mod p
   b1 ← +(C^((q+1)/4)) mod q
   b2 ← −(C^((q+1)/4)) mod q
   // The algorithm for the Chinese remainder algorithm is called four times.
   P1 ← Chinese_Remainder (a1, b1, p, q)
   P2 ← Chinese_Remainder (a1, b2, p, q)
   P3 ← Chinese_Remainder (a2, b1, p, q)
   P4 ← Chinese_Remainder (a2, b2, p, q)
   return P1, P2, P3, and P4
}
```

Rabin Cryptosystem…

- Example

1. Bob selects p = 23 and q = 7. Note that both are congruent to 3 mod 4.
2. Bob calculates $n = p \times q = 161$.
3. Bob announces n publicly; he keeps p and q private.
4. Alice wants to send the plaintext P = 24. Note that 161 and 24 are relatively prime; 24 is in Z_{161}^*. She calculates $C = 24^2 = 93$ mod 161, and sends the ciphertext 93 to Bob.

Rabin Cryptosystem…

- Example…

5. Bob receives 93 and calculates four values:

   $a_1 = +(93^{(23+1)/4}) \bmod 23 = 1 \bmod 23$

   $a_2 = -(93^{(23+1)/4}) \bmod 23 = 22 \bmod 23$

   $b_1 = +(93^{(7+1)/4}) \bmod 7 = 4 \bmod 7$

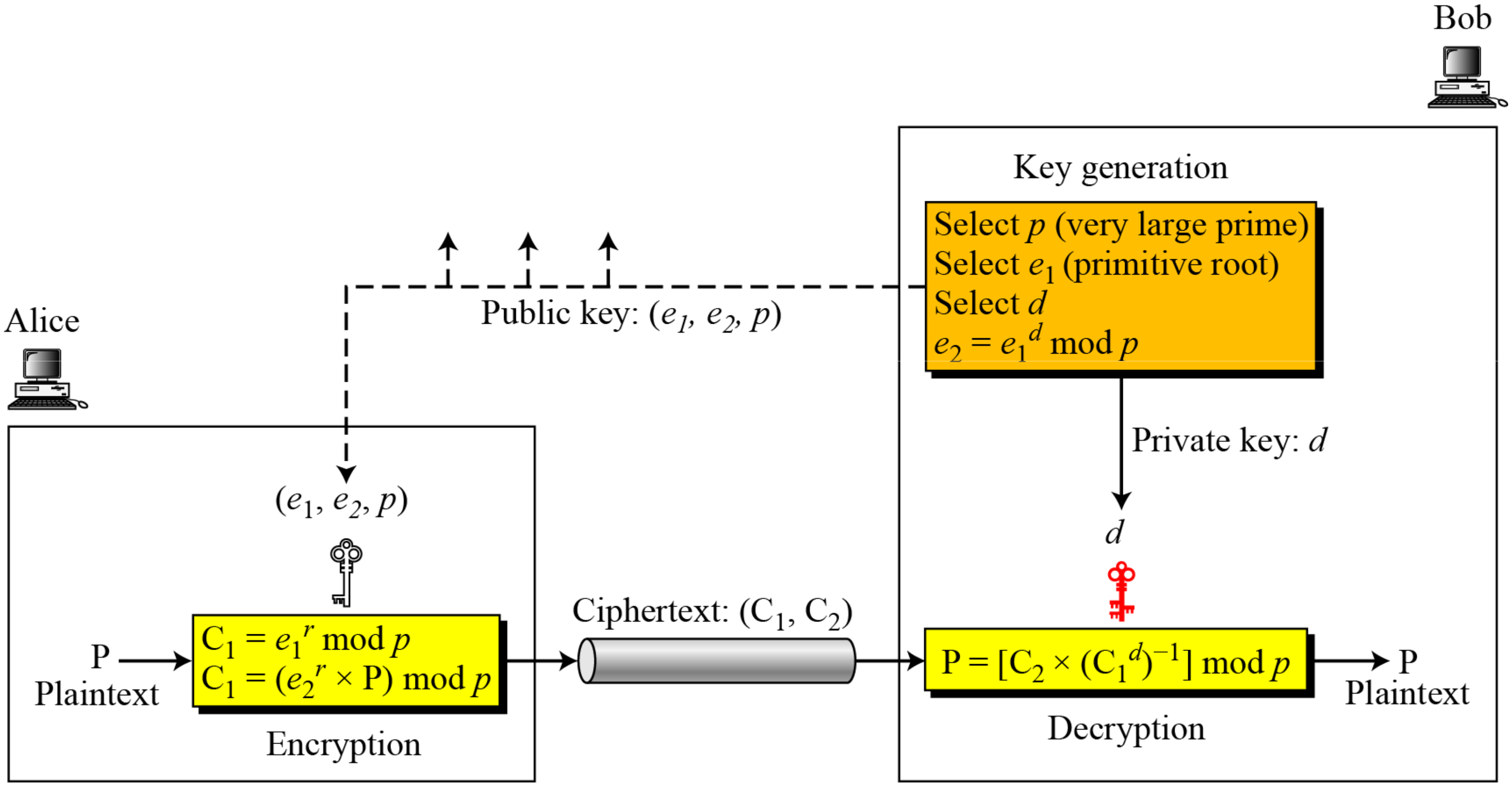
   $b_2 = -(93^{(7+1)/4}) \bmod 7 = 3 \bmod 7$

6. Bob takes four possible answers, (a_1, b_1), (a_1, b_2), (a_2, b_1), and (a_2, b_2), and uses the Chinese remainder theorem to find four possible plaintexts: 116, 24, 137, and 45. Note that only the second answer is Alice's plaintext.

El-Gamal Cryptosystem

- Besides RSA and Rabin, another public-key cryptosystem is ElGamal.
- ElGamal is based on the discrete logarithm problem

El-Gamal Cryptosystem…



El-Gamal Cryptosystem...

- Key Generation

Algorithm 10.9 *ElGamal key generation*

```
ElGamal_Key_Generation
{
    Select a large prime p
    Select d to be a member of the group G = <Zp*, ×> such that 1 ≤ d ≤ p − 2
    Select e1 to be a primitive root in the group G = <Zp*, ×>
    e2 ← e1^d mod p
    Public_key ← (e1, e2, p)                // To be announced publicly
    Private_key ← d                         // To be kept secret
    return Public_key and Private_key
}
```

El-Gamal Cryptosystem…

- Encryption

Algorithm 10.10 *ElGamal encryption*

ElGamal_Encryption $(e_1, e_2, p, \mathrm{P})$ // P is the plaintext

{

Select a random integer r in the group $\mathbf{G} = < \mathbf{Z}_p{}^*, \times >$

$\mathrm{C}_1 \leftarrow e_1{}^r \bmod p$

$\mathrm{C}_2 \leftarrow (\mathrm{P} \times e_2{}^r) \bmod p$ // C_1 and C_2 are the ciphertexts

return C_1 and C_2

}

El-Gamal Cryptosystem...

- Decryption

Algorithm 10.11 *ElGamal decryption*

```
ElGamal_Decryption (d, p, C1, C2)           // C1 and C2 are the ciphertexts
{
    P ← [C2 (C1^d) ^-1] mod p               // P is the plaintext
    return P
}
```

The bit-operation complexity of encryption or decryption in ElGamal cryptosystem is polynomial.

El-Gamal Cryptosystem...

- Example
  - Here is a trivial example. Bob chooses p = 11 and e_1 = 2 and d = 3.
  - $e_2 = e_1^d = 8$. So the public keys are (2, 8, 11) and the private key is 3.
  - Alice chooses r = 4 and calculates C_1 and C_2 for the plaintext 7.

Plaintext: 7

$C_1 = e_1^r \bmod 11 = 16 \bmod 11 = 5 \bmod 11$

$C_2 = (P \times e_2^r) \bmod 11 = (7 \times 4096) \bmod 11 = 6 \bmod 11$

Ciphertext: (5, 6)

El-Gamal Cryptosystem…

- Example…
  - Bob receives the ciphertexts (5 and 6) and calculates the plaintext.

$$[C_2 \times (C_1{}^d)^{-1}] \bmod 11 = 6 \times (5^3)^{-1} \bmod 11 = 6 \times 3 \bmod 11 = 7 \bmod 11$$

Plaintext: 7

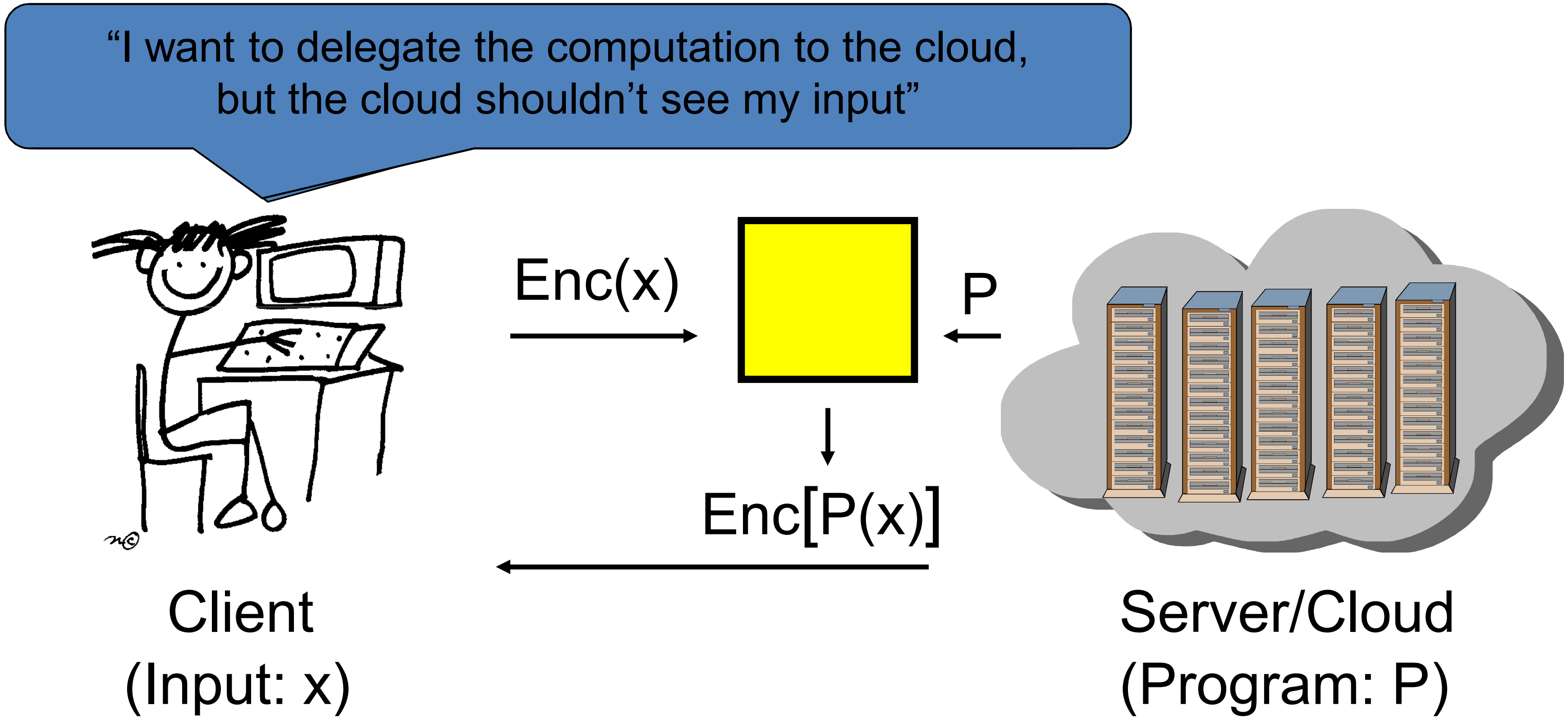
El-Gamal Cryptosystem...

- Example...
  - Instead of using $P = [C_2 \times (C_1^{d})^{-1}] \bmod p$ for decryption, we can avoid the calculation of multiplicative inverse and use $P = [C_2 \times C_1^{p-1-d}] \bmod p$.
  - We can calculate $P = [6 \times 5^{11-1-3}] \bmod 11 = 7 \bmod 11$.
  - (Apply Fermat's little theorem $a^{-1} \bmod p = a^{p-2} \bmod p$)

For the ElGamal cryptosystem, *p* must be at least 300 digits and *r* must be new for each encipherment.

HOMOMORPHIC ENCRYPTION

The basic idea: Computing on encrypted data



Homomorphic encryption

- a form of encryption which allows specific types of computations to be carried out on ciphertext and generate an encrypted result which, when decrypted, matches the result of operations performed on the plaintext.
- allow the chaining together of different services without exposing the data to each of those services

Homomorphic encryption

- Partially homomorphic encryption schemes
  - Either addition [e.g. Paillier] or
  - Multiplication [e.g. Elgamal]
- Fully homomorphic encryption schemes
  - Both addition and multiplication [e.g. Gentry et al.]

Homomorphic encryption

- Partially homomorphic encryption schemes
  - Malleable by design
  - An encryption algorithm is malleable if it is possible for an adversary to transform a ciphertext into another ciphertext which decrypts to a related plaintext.
  - That is, given an encryption of a plaintext , it is possible to generate another ciphertext which decrypts to , for a known function , without necessarily knowing or learning .

Assignment questions

1. BOB chooses p=101, q=113 and therefore n=11413.

   $\phi(n)=11200=2^6 \times 5^2 \times 7$

   Can the following be candidates of e?

   a. 25

   b. 32

   Justify your answer.

Assignment questions…

2. Given (e,n), one would not be able to find d. Proove.
3. "If the value of d is leaked, then changing it is not suffice. One needs to change the modulus n." Comment on the statement.

Assignment questions…

4. Comment on the homomorphic property of RSA
5. In an unpadded RSA cryptosystem, a plaintext m is encrypted as $E(m)=m^e \bmod n$, where (e,n) is the public key. Given such a ciphertext, can an adversary construct an encryption of mt for any integer t?

Now, think about the case when RSA is used with OAEP.